PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+incremental +backup, +file +allocation full backup, partial ba SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
incremental backup file allocation full backup partial backup archive restore single volume

Found **222** of **151,219**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale □ ▨ ▨ ▨ ■

1 ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging ■
C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, Peter Schwarz
March 1992 **ACM Transactions on Database Systems (TODS)**, Volume 17 Issue 1

Full text available: pdf(5.23 MB) Additional Information: full citation, abstract, references, citings, index terms, review

DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

Keywords: buffer management, latching, locking, space management, write-ahead logging

2 Recovery Techniques for Database Systems ■
Joost S. M. Verhofstad
June 1978 **ACM Computing Surveys (CSUR)**, Volume 10 Issue 2

Full text available: pdf(2.32 MB) Additional Information: full citation, references, citings, index terms

3 File servers for network-based distributed systems ■
Liba Svobodova
December 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 4

Full text available: pdf(4.23 MB) Additional Information: full citation, references, citings, index terms, review

4 An efficient and flexible method for archiving a data base ■
C. Mohan, Inderpal Narang
June 1993 **ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD international conference on Management of data**, Volume 22 Issue 2

Full text available: pdf(969.13 KB) Additional Information: full citation, abstract, references, citings, index terms

We describe an efficient method for supporting incremental and full archiving of data bases (e.g., individual files). Customers archive their data bases quite frequently to minimize the duration of data outage. Because of the growing sizes of data bases and the ever increasing need for high availability of data, the efficiency of the archive copy utility is very important. The method presented here minimizes interferences with concurrent transactions by not acquiring any locks on the data b ...

5 A structural view of the Cedar programming environment
Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann
August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

6 Compiler transformations for high-performance computing
David F. Bacon, Susan L. Graham, Oliver J. Sharp
December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms, review

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

7 The Alpine file system
M. R. Brown, K. N. Kolling, E. A. Taft
November 1985 **ACM Transactions on Computer Systems (TOCS)**, Volume 3 Issue 4

Full text available: pdf(2.95 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Alpine is a file system that supports atomic transactions and is designed to operate as a service on a computer network. Alpine's primary purpose is to store files that represent databases. An important secondary goal is to store ordinary files representing documents, program modules, and the like. Unlike other file servers described in the literature, Alpine uses a log-based technique to implement atomic file update. Another unusual aspect of Alpine is that it performs all commu ...

8 Petal: distributed virtual disks
Edward K. Lee, Chandramohan A. Thekkath
September 1996 **Proceedings of the seventh international conference on Architectural**

support for programming languages and operating systems, Volume 31 , 30 Issue 9 , 5

Full text available: pdf(1.10 MB) Additional Information: full citation, abstract, references, citings, index terms

The ideal storage system is globally accessible, always available, provides unlimited performance and capacity for a large number of clients, and requires no management. This paper describes the design, implementation, and performance of Petal, a system that attempts to approximate this ideal in practice through a novel combination of features. Petal consists of a collection of network-connected servers that cooperatively manage a pool of physical disks. To a Petal client, this collection appear ...

9 The HP AutoRAID hierarchical storage system
John Wilkes, Richard Golding, Carl Staelin, Tim Sullivan
February 1996 **ACM Transactions on Computer Systems (TOCS)**, Volume 14 Issue 1

Full text available: pdf(1.82 MB) Additional Information: full citation, abstract, references, citings, index terms

Configuring redundant disk arrays is a black art. To configure an array properly, a system administrator must understand the details of both the array and the workload it will support. Incorrect understanding of either, or changes in the workload over time, can lead to poor performance. We present a solution to this problem: a two-level storage hierarchy implemented inside a single disk-array controller. In the upper level of this hierarchy, two copies of active data are stored to provide f ...

Keywords: RAID, disk array, storage hierarchy

10 Managing energy and server resources in hosting centers
Jeffrey S. Chase, Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat, Ronald P. Doyle
October 2001 **ACM SIGOPS Operating Systems Review , Proceedings of the eighteenth ACM symposium on Operating systems principles**, Volume 35 Issue 5

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings, index terms

Internet hosting centers serve multiple service sites from a common hardware base. This paper presents the design and implementation of an architecture for resource management in a hosting center operating system, with an emphasis on *energy* as a driving resource management issue for large server clusters. The goals are to provision server resources for co-hosted services in a way that automatically adapts to offered load, improve the energy efficiency of server clusters by dynamically res ...

11 Industry perspectives: Toward autonomic computing with DB2 universal database
Sam S. Lightstone, Guy Lohman, Danny Zilio
September 2002 **ACM SIGMOD Record**, Volume 31 Issue 3

Full text available: pdf(785.28 KB) Additional Information: full citation, abstract, references

As the cost of both hardware and software falls due to technological advancements and economies of scale, the cost of ownership for database applications is increasingly dominated by the cost of people to manage them. Databases are growing rapidly in scale and complexity, while skilled database administrators (DBAs) are becoming rarer and more expensive. This paper describes the self-managing or autonomic technology in IBM's DB2 Universal Database® for UNIX and Windows to illustrate how self ...

12 IS '97: model curriculum and guidelines for undergraduate degree programs in information systems
Gordon B. Davis, John T. Gorgone, J. Daniel Couger, David L. Feinstein, Herbert E.

Longenecker
December 1996 **ACM SIGMIS Database , Guidelines for undergraduate degree programs on Model curriculum and guidelines for undergraduate degree programs in information systems**, Volume 28 Issue 1
Full text available: pdf(7.24 MB) Additional Information: full citation, citings

13 Scale and performance in a distributed file system
John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, Michael J. West
February 1988 **ACM Transactions on Computer Systems (TOCS)**, Volume 6 Issue 1
Full text available: pdf(2.38 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The Andrew File System is a location-transparent distributed tile system that will eventually span more than 5000 workstations at Carnegie Mellon University. Large scale affects performance and complicates system operation. In this paper we present observations of a prototype implementation, motivate changes in the areas of cache validation, server process structure, name translation, and low-level storage representation, and quantitatively demonstrate Andrews ability to scale gracefully. W ...

14 Query evaluation techniques for large databases
Goetz Graefe
June 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 2
Full text available: pdf(9.37 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Database management systems will continue to manage large data volumes. Thus, efficient algorithms for accessing and manipulating large sets and sequences will be required to provide acceptable performance. The advent of object-oriented and extensible database systems will not solve this problem. On the contrary, modern data models exacerbate the problem: In order to manipulate large sets of complex objects as efficiently as today's database systems manipulate simple records, query-processi ...

Keywords: complex query evaluation plans, dynamic query evaluation plans, extensible database systems, iterators, object-oriented database systems, operator model of parallelization, parallel algorithms, relational database systems, set-matching algorithms, sort-hash duality

15 Compactly encoding unstructured inputs with differential compression
Miklos Ajtai, Randal Burns, Ronald Fagin, Darrell D. E. Long, Larry Stockmeyer
May 2002 **Journal of the ACM (JACM)**, Volume 49 Issue 3
Full text available: pdf(348.32 KB) Additional Information: full citation, abstract, references, citings, index terms

The subject of this article is *differential compression*, the algorithmic task of finding common strings between versions of data and using them to encode one version compactly by describing it as a set of changes from its companion. A main goal of this work is to present new differencing algorithms that (i) operate at a fine granularity (the atomic unit of change), (ii) make no assumptions about the format or alignment of input data, and (iii) in practice use linear time, use constant spa ...

Keywords: Delta compression, differencing, differential compression

16 A system software architecture for high-end computing
David S. Greenberg, Ron Brightwell, Lee Ann Fisk, Arthur Maccabe, Rolf Riesen
November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(254.54 KB) Additional Information: full citation, abstract, references, citings

Large MPP systems can neither solve grand-challenge scientific problems nor enable large scale industrial and governmental simulations if they rely on extensions to workstation system software. At Sandia National Laboratories we have developed, with our vendors, a new system architecture for high-end computing. Highest performance is achieved by providing applications with a light-weight interface to a collection of processing nodes. Usability is provided by creating node partitions specialized ...

17 Distributed file systems: concepts and examples
Eliezer Levy, Abraham Silberschatz
December 1990 **ACM Computing Surveys (CSUR)**, Volume 22 Issue 4

Full text available: pdf(5.33 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The purpose of a distributed file system (DFS) is to allow users of physically distributed computers to share data and storage resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). A DFS is implemented as part of the operating system of each of the connected computers. This paper establishes a viewpoint that emphasizes the dispersed structure and decentralization of both data and con ...

18 Translator writing systems
Jerome Feldman, David Gries
February 1968 **Communications of the ACM**, Volume 11 Issue 2

Full text available: pdf(4.47 MB) Additional Information: full citation, abstract, references, citings

A critical review of recent efforts to automate the writing of translators of programming languages is presented. The formal study of syntax and its application to translator writing are discussed in Section II. Various approaches to automating the postsyntactic (semantic) aspects of translator writing are discussed in Section III, and several related topics in Section IV.

Keywords: compiler compiler-compiler, generator, macroprocessor, meta-assembler, metacompiler, parser, semantics, syntactic analysis, syntax, syntax-directed, translator, translator writing system

19 Tar and Taper for Linux: Learn to use tar and the friendly taper archival tools.
Yusuf Nagree
February 1996 **Linux Journal**

Full text available: html(33.29 KB) Additional Information: full citation, index terms

20 The evolution of the DECsystem 10
C. G. Bell, A. Kotok, T. N. Hastings, R. Hill
January 1978 **Communications of the ACM**, Volume 21 Issue 1

Full text available: pdf(1.92 MB) Additional Information: full citation, abstract, references, citings, index terms

The DECsystem 10, also known as the PDP-10, evolved from the PDP-6 (circa 1963) over five generations of implementations to presently include systems covering a price range of five to one. The origin and evolution of the hardware, operating system, and languages are described in terms of technological change, user requirements, and user developments. The PDP-10's contributions to computing technology include: accelerating the transition from batch oriented to time sharing computing systems; ...

Keywords: architecture, computer structures, operating system, timesharing

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	5	(Skiba near mark).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:29
L2	2	(Ryzhkin near Mikhail).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:30
L3	5	1 or 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:30
L4	520	incremen$4 near2 (backup or (back$4 adj up) or archiv$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:32
L5	315050	restor$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:32
L6	254842	operat$4 adj system	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:32
L7	9789	file adj (management or allocation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:33
L8	1253	single adj volume	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:33
L9	311	4 and 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:33

L10	200	9 and 6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:33
L11	31	10 and 7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:34
L12	0	11 and 8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/03/29 14:33